

PROCESSES SYSTEMS AND NETWORKS FOR SECURED
INFORMATION EXCHANGE USING COMPUTER HARDWARE

Background of the Invention

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of co-pending U.S. patent applications, Ser. No. 09/198,411 filed 24 November 1998 in the name of the same inventor and entitled PROCESSES AND SYSTEMS FOR SECURED INFORMATION EXCHANGE USING COMPUTER HARDWARE; Ser. No. 09/241,860 filed 2 February 1999 in the name of the same inventor and entitled GENERAL PURPOSE ACTIVE-BACKPLANE AND EXPANSION-BUS COMPATIBLE SINGLE BOARD COMPUTER FOR SECURED INFORMATION EXCHANGE, and Ser. No. 09/241,861 filed 2 February 1999 in the name of the same inventor and entitled PROCESSES AND SYSTEMS FOR SECURED INFORMATION EXCHANGE USING COMPUTER HARDWARE.

Field of the Invention

This invention relates to methods and devices for the safe and secure operation of host information systems which must exchange information with other information systems and devices, such as in cyberspace and, where such external systems may be corrupted in some manner, utilizing system architecture and data signal transformations as opposed to conventional software based firewalls to receive and convert or reformat incoming information signals from the external

systems and thereafter extracting and supplying only non corrupted information signals to the host systems. The invention also provides for screening of outgoing information signals from the host systems to prevent unauthorized information exchange and for permitting secure updating of host systems files with information before updated files are returned to the host systems. Additionally, the invention provides a suitable platform for adaptive-management of systems and networks. A framework for such adaptive capabilities is provided in "An Adaptive Reference Monitor for Information System Security ;" IEEE Proceedings 1st Annual Conference on Neural Networks '87. C Kellum. This document is referenced briefly in the body of this application.

Description of the Prior Art

The field of information-system security (InfoSec) technology and practice to date has focused on controlling human user access to computer system resources, and preventing hostile, clandestine computer programs, such as computer viruses, from corrupting a computer system. The advent of the Internet and personal computers brought new challenges to the InfoSec field, particularly because in networks, other machines, not human users, were the entities that primarily accessed a computer system. Old, pre-network, password usage and similar software authentication methods only offered a modicum of security control at *authorized user* entry points of a network. Intruders could bypass these methods as

they do in today's Internet and tap or hack (i.e. the term hackers) into the communications segment of a computer network and launch any form of mischief or disruption that the target network would allow. This is the core of today's Internet security problem, wherein intruders can disrupt nearly all forms of Internet activity, from disabling web sites and compromising message traffic, to falsifying identity. The conventional InfoSec problems of unauthorized user access, incorrect operation, and system malfunction remain, in addition to today's network oriented security problems.

Various schemes of varying degrees of complexity and convolution have been devised to provide needed security. Examples of two of the latest of such schemes are U.S. Patents 5,623,601 to Vu, and 5,632,011 to Landfield, et al. The methods taught are implemented as software computer programs, which operate with or as a standard operating system software package. Assumed in the methods are the correct implementation and operation of these software packages, and the operating system (i.e. control software) with which it must operate. Here, "correct operation" also includes InfoSec correctness which means no compromise to a hosting system is precipitated by the operation of such software. Proving or verifying such assertions as software correctness, or software operational integrity remains a major barrier in InfoSec technology, as well as in computer science and engineering in general. Software verification is a formidable undertaking. Finally, software (i.e. computer programs) is vulnerable to compromise by other computer programs, which may include viruses. Software attack and corruption, whether e-

mail packages, protocol modules, operating systems, macro services such as OPEN commands, etc. is the realm of the system/network intruder (the Hacker). The ideal InfoSec tool should not be software dependant.

Today's InfoSec tools such as the above cited references implement, in software, a type of gateway function. The term firewall is often used. A gateway is a computer that connects two different networks together. A firewall is a gateway with the additional constraints and properties that all inter-network traffic must pass through it, whereby all unauthorized (according to some rule-set or security policy) traffic is prevented from passage. The firewall must operate correctly and be free from compromise. To further compound this difficulty, firewalls are filters. As such they must allow selected external traffic to pass through to the system or network being protected, especially if useful information exchange between the systems and networks separated by the firewall, is to take place. Firewalls have no way to filter out hostile traffic, without prior knowledge of such traffic. Also, service packages, such as e-mail, containing corrupted command macro programs (e.g. macro viruses) are impervious to firewalls. Possible legitimate bit configurations in command fields of standard message traffic passing through a firewall could trigger disruptive events, when entering a protected system or network. Firewalls, acting as an address translation proxy for an inside/protected system or network, can protect that system or network from exposure, to an external system or network, of its internal and critical address information. Again, one assumes (usually, without rigorous basis) correctness of the proxy software function.

Although firewalls and anti-virus software were steps in the right direction, more universal protection of information systems or networks is needed, whereby such protection is easily verifiable, cost-effective, and does not require "apriori knowledge" to successfully execute a detection and/or filtering function, and is software independent. As an example, the multi-level security (MLS) problem is generally thought to be unsolvable. Briefly stated, the MLS problem is how to enable users of different (and possibly hierarchal) security classes, share the same network resources, while preventing unauthorized information transfer. Old solution attempts focused on secure labeling, with its inherent problems of integrity, increased storage, verification of label processing, etc. Further, due to the lack of available "trusted-computer-base" (TCB) technology & systems, a process consisting of aggregate labeled processes must assume the label of the highest level of the aggregate. This is often referred to as the "security creep" problem.

Summary of the Invention

The present invention is directed to the use of a computer hardware device which functions as an inter domain screen or signal processor hereafter referred to as the IDS. The IDS is a unique data flow control architecture and device family, within which two unique processes are executed. The IDS protects its host system from compromise from any external connections. The IDS contains an intermediate-domain-device (IDD), sockets which connect the IDD to the host system, and sockets

which connect the IDD to external domains. External domains, which are to exchange information with the host, are prevented, by the IDS from compromising the host. The intermediate domain (embodied by the IDD) is a special purpose domain for information exchange. The purpose of the IDS is to permit maximum information interchange, while preventing external signals from directly entering a protected domain or host. The term "host system" is used synonymously with "protected domain". The external signals may be the carrier of hostile executable code. Viruses, worms, triggers for trap-door and Trojan horse type software, and other forms of hostile signals use incoming data signals to enter a protected (target) information system environment. That is, the information being exchanged, including any hostile data, is contained in data sets carried by signals. The hostile data sets depend on the structural integrity of the incoming data stream or signal(s) for the necessary maintenance of its own structure. With the present invention, this structural integrity is disrupted, while the information carried by the data stream is preserved in the IDS. The InfoSec processes executed are isolation of external signals, and derivation of the information content of such signals and are referenced as a modified-read process. To achieve this, an "information-preserving" data transformation takes place in the IDS on these potentially corrupted incoming external data signals such as by processing an incoming signal containing an initial data set in such a manner as to extract the information in the initial data set, thus creating a signal having a different data set, and, thereafter, transmitting the different data sets to the host domain. Such processing includes converting the type and/or

format of signals such as converting a telephone signal to a T.V. signal or converting an analog signal to a digital signal.

The intermediate domain and the modified-read function which takes place therein form a protective screen for the internal or host system or domain, to which they are attached. The modified-read process does not require prior knowledge of a particular virus/worm, etc and is a universal eliminator of hostile executable code.

The IDS therefore is not a proxy-server or firewall which are vulnerable to software errors and/or compromise, and to unknown hostile executable code (i.e. new virus) penetration. The IDS is an incoming signal buffer and transformer and an outgoing signal filter. It is a hardware device that is scalable, that provides the special purpose domain for information data flow control. This special purpose domain is intermediate between the IDS's host system, which it is protecting, and external systems.

It is important to note that generic IDS functions and architecture enforce the following for the systems/networks it is protecting:

- a) immunity to penetration;
- b) assurance that all traffic between the protected domain and the external domain enters the IDS;
- c) no direct connections between the protected domain the external domain exist; and
- d) only authorized information, as defined by local InfoSec policy is allowed to exit the IDS.

The IDS is a multi-function device acting as a firewall, a guard/filter, a network front-end, and hostile code (e.g. virus) eliminator. The IDS may also act as a host system file screen which is adapted to receive file information from the host system, screen new file information and thereafter update existing files in the host system.

The present invention is directed to a method and apparatus for enabling information to be exchanged between a protected system and an external information source wherein the information is contained in data sets which are carried by signals in such a manner that undesired data is prevented from reaching the protected system. The invention uses an intermediate domain computer hardware device which is connected between the external data source and the protected system so as to receive an initial data set including the information which may contain undesirable data transmitted from the external source. In the intermediate domain hardware, the signals containing the initial data set are processed to create a second data set in such a manner that the information in the initial data set, is extracted to thereby screen out undesirable data. Thereafter, the extracted information is passed to the protected system.

In the elementary version of the invention, the intermediate domain computer hardware device (the IDD) may be a network computer, a webtv unit, a single board computer (SBC), a laptop/notebook computer, other personal computer (or the like), or a specially designed chip which receives signals in any manner such as broadcast signals or signals from a conventional telephone line from an external domain site such as the Internet or world wide web. The incoming signals to the hardware

device are routed (via the IDD) to what is tantamount to a tv card associated with a PCI bus of a computer system. The intermediate/domain device (IDD) in the form of the webtv system transforms the incoming signals. Any virus contained in the original signals can not survive the transformation of the signal format from the signals originally received, (such as by way of the telephone line), to the video signals at the tv-card and thus the card supplies extracted information to a connected computer which may be a personal computer.

As a reduced function (manual) embodiment of the elementary system, signals from the computer hardware device such as in a webtv system may be conveyed to a conventional printer wherein the signals are converted or transformed into a printed format which may be preserved. By taking the printed format and transforming the printed format into signals, such as by scanning, the information can be provided from the printer to the input of a PC such that only the extracted information without any viruses is passed to the personal computer or host system. In both the foregoing scenarios, the virus (or other forms of hostile code) can not survive the signal transformation within the intermediate domain system.

As mentioned, as opposed to using the webtv unit, a single board computer, laptop or notebook computer may be utilized to act as the intermediate domain device. The laptop or single board computer is connected to receive a signal such as through a telephone line from the Internet. The invention, however, is not limited in its application to single point or individual host or host systems. the host may be

substantially any single receiving information processor including main frame computers, information networks including local and wide area networks (LANs and WANs) and the like. Also, the computer hardware of the IDS is not limited to single or individual computer elements but may be computer networks and systems.

Any contamination of the IDS' intermediate domain from system error or hostile executable code from external domains, is easily corrected by a reset function, or a cold-boot from a clean boot-disk. For some applications, this could be a recommended periodic procedure. The IDS architecture insures that only data that has gone through a modified-read process enters the host (protected) system.

In another embodiment of the invention, the IDS is used to safely update files stored in the host system. In this embodiment, a file from the host is loaded to the IDS. The IDS also receives information signals from the external domain and processes the signals in a modified-read to thereby convert the signal to change an initial data set to a second data set in such a manner as to extract the information from the second data set and updates the file loaded from the host and thereafter forwards the safely updated file to the host. In this manner all updating of files is done in a manner in which the host files can not be compromised.

In yet another embodiment, the invention provides a screening of all outgoing signals from the host or protected system to the IDS so as to ensure that only permitted information is transmitted.

It is the primary object of the present invention to provide a method and apparatus which protects a host system from contamination by preventing external

signals from entering the protected host system permitting safe "information" exchange between the host and possibly hostile external domains and, in some embodiments, also preventing inadvertent and/or unauthorized release of data from the host system.

These and additional capabilities, utility, and attainments of the present invention, should become apparent to those skilled in the art, upon reading of the following detailed description when taken in conjunction with the drawings wherein there is shown and described illustrative embodiments of the invention.

Brief Description of the Drawings

In the course of the following detailed description, reference will be made to the attached drawings in which:

Fig. 1 is an illustration of a prior art firewall configuration wherein a protected system is connected to an external system via an intervening firewall arrangement consisting of a gateway function processor surrounded on either side by a router function;

Fig. 2 illustrates an intermediate domain screen (**IDS**) device of the present invention separating an internal or host domain, that is protected by the invention, and an external domain that might be hostile and/or corrupted;

Fig. 3 illustrates an arrangement of several IDS devices in accordance with the invention, each of which use an authentication process for mutual identification, thus

forming a secure network overlaying an intervening public or unprotected network;

Fig. 4 illustrates another embodiment of the invention, wherein an IDS is configured to protect several internal domains from corruption or compromise by an external domain;

Fig. 5 diagrams the modified-read process executed by the invention of Fig. 2;

Fig. 6 illustrates a basic or elementary version of the invention;

Fig. 7 illustrates the generic logic structure of the invention;

Fig. 8 illustrates a multi-function embodiment of the invention;

Fig. 9 illustrates a modification to a single board computer functioning as an intermediate domain device internal to its' host;

Fig. 10 illustrates an InfoSec related signal flow control feature;

Fig. 11 illustrates a network where sites/nodes protected by the invention (the IDS device), form a secure subnetwork;

Fig. 12 illustrates the generic architecture of the invention;

Fig. 13 illustrates an example system-on-chip (SOC) version of the invention.

Detailed Description of the Invention

The invention has several fundamental embodiments which are described in the following sections. Other embodiments are derived from these fundamental embodiments. The term "domain" is used throughout this document. "Domain" is defined as a system or network or set of systems or networks. The term "router"

refers to a computer that selects and implements, at the software level, data-paths from one location to another in a computer network. Also the term "signal" is used synonymously with data, data sets, files, messages, packets, protocol sequences, etc. throughout this document, to stress generality. Signals, as referenced herein, refer to any information carrying quanta, such as electro-magnetic current, lightwaves, which are processable by information system technology. It is fundamental to realize that data, data sets, control commands, etc., are manifested as electronic signals and/or electro-optic signals and that information systems and networks transform and transceive such signals, and that the invention as described more fully below, operates at this fundamental signal level.

Prior Art Attempts

Referring to Fig. 1, there is illustrated a prior art firewall arrangement. An ordinary gateway function module 1 sits between two filtering routers 3 and 4. One router 3 is connected to an internal network 5 and the gateway 1. The other router 4 is connected to an external network 6 and the gateway. These modules and especially their software must interact in an error-free and complex fashion to enforce a security policy for information transfer between the internal network and the external network. Since these modules primarily implement a filtering function 2, which implies that externally generated signal traffic will enter the internal network. Such traffic may be contaminated, and thus compromise the internal network. All

methods in current practice are software based, and operate on a framework derivable from that depicted in Fig. 1. Generally, software cannot be "trusted" to function correctly, where "trusted" is defined to include provable correctness in structure, compilation, installation, operation. Also hacking and other types of intrusions attack the software of the networks that are targeted. A prime example is the Internet where intrusions, hacking, web-site compromise, and other forms of software misuse are rampant.

Hardware-Based InfoSec Provided by the Present Invention

Referencing Fig. 2, the intermediate domain screen (IDS) 10 of the present invention is a hardware system composed at least three (3) and in some embodiments of four (4) generic hardware components. The basic components are an Intermediate-Domain-Device (IDD) 12, an external domain socket 14, and an IDS to internal domain socket 13. A fourth hardware component is an internal domain to IDS socket 17. The sockets can take the form of conventional modem type devices including special purpose signal processing and signal transfer components such as video, wireless communication, integrated telephony, and facsimile cards and the like, programmable systems or devices such as single board computers (SBC), smart digital signal processors, embedded systems and the like, large mainframes, local and/or wide area networks (LANs/WANs). The invention physically and logically separates an internal domain 15 from an external domain 16. The internal

domain can range from a single system such as a personal computer or web site to a network, as can the external domain. The internal domain is the domain being protected by the invention, and is referred to as the host or protected domain. Each of the sockets 13, 14 and 17 can be implemented as a set of sockets. Socket 13 allows only specific types of signals or data sets to enter the host 15. Socket 17 performs a filter or guard function between the host 15 and the IDS, to restrict and control the release of signals from the host 15. The IDD 12, acts as a confinement domain for external signals or data sets carried by incoming signals, thus preventing viruses (and other forms of hostile code) contained in the external signals from entering the protected domain or host. The IDD can range in size and functionality (as defined in applications cited in the cross reference section above) from a single chip or card, to a special purpose device, to a network, to a large mainframe. The IDD provides an intermediate domain for safe information interchange between the internal-domain/host 15 and the external domain 16. This interchange includes execution of external programs, Internet access such as web browsing, updating internal-domain programs and software, which have been sent, via socket 17, to the IDD by a host filtering or selection process residing in the IDD for updating and/or other interaction with the external domain. The IDD executes an "information-preserving-data-transformation" process to extract necessary information from external signals and transmits such information, via socket 13, to the host 15. This process is called a modified-read (M-R), and in conjunction with socket 13 insures that only uncontaminated signals or data sets are transmitted to the host. Socket 13

transmits only signals that have undergone the (M-R) process.

The socket components 14, 13 and 17 must not communicate directly with each other in an IDS configuration. This could facilitate unauthorized data transfers. All data transfer must be monitored by the IDD 12. As an example; to insure this, a bus request pin of a network interface card, NIC, embodying socket 14 must be deactivated, i.e. grounded. This results in a "partially connected" hardware architecture. In many instances, InfoSec concerns must also include the possibility of compromise from within. Such compromise can be malicious, or inadvertent. The inadvertent compromise can result from system malfunction and/or user/operator error. In the data flow control framework, the unauthorized release of information as a result of such compromise is addressed by the invention, wherein the IDS can restrict host 15 users, i.e. insiders, to specific, controlled functions relative to the external domain 16. Socket 17 operation supplies a filter or guard function, the purpose of which is to prevent unauthorized release of data or information from a protected host. In this respect, the socket 17 may include a single board computer which is programmable to filter or screen signals passing from the host to the IDD so that only authorized or releasable data is allowed to enter the IDD from the host.

Large environments, such as networks are typical applications for versions of the IDS. Thus advanced, sophisticated filtering type functions can be implemented. Depending on the processing power of the component chosen to implement the IDD 12, the filter function can range from a simple template-matching query filter to

highly sophisticated, adaptive, cognitive, content analyzing, auto-classifier type capabilities. As a hardware system, the IDS 10 physically separates its host computer systems from an external system or network at the signal level. Thus, all viruses, worms, and other forms of hostile executable code contained in external signals or data sets are prevented from entering the host system, because all external signals are confined to the IDD 12. In a standard IDS session, the IDS receives data, some of which might possibly be contaminated from external domains, extracts the "information" contained in this data, and safely transmits such "information" to the protected host 15. The IDS can then purge the (previously received) external signals and reset (e.g. cold boot) itself to an uncontaminated initial state. This purge-reset function, of the IDS, can be automatic and/or user activated.

The "per session" auto decontamination process of the IDS, is an F-T feature of the IDS. At start-up, shutdown, and per session flush of the IDS accomplishes the FLUSH function. The IDS, thus, always enters a session in a decontaminated state. This is possible because the flush doesn't cause any loss of needed information, since the IDS is a "communications front-end" processor. As such, it is operationally expedient for an IDS to be "flushed" such that each session starts in a "clean" state. Therefore, a network of IDS's can auto-eliminate/decontaminate itself, using its signal confinement and flush features. Additionally, information signals transmitted by the IDS have undergone an M-R process. Thus, the propagation of contamination (e.g. worm functions) is prevented. Consequently, an IDS cleans a

network of contamination, and maintains a contaminate-free network.

Basic versions of the IDS implement a video-transformation modified-read process. This is a signal level (information preserving) data transformation. No outbound data or signal path from the host system exists. Thus unauthorized clandestine or inadvertent transmission of host data, is prevented. In the programmable IDS versions where signals are transmitted from the host, a comprehensive generic processor-based intermediate domain is provided which can be used with smart adaptive InfoSec agent programs capable of hostile-penetration countermeasure type functions. These functions include adaptive classifiers, session encryptors, and e-mail (payload) encryption functions, for safe transit of outgoing IDS data.

All IDS versions can also reside remotely from their host system. Such versions can be configured to protect several host systems simultaneously. The IDS architecture easily accommodates IDS to host encryption (i.e. end-to-end encryption) to protect data in transit through public networks linking the host and the IDS. Hybrid versions of the IDS which implement a modified-read (M-R) function to remove hostile data from incoming data streams, simultaneously implement a filter function, to prevent unauthorized data exfiltration from the host. The hybrid version combines any set of IDS versions to screen incoming traffic and outgoing traffic. It thus allows the host safe and simultaneous connectivity to domains of different security levels. In addition, the IDD, intermediate domain device can be set to control the host systems. In this mode of operation, the IDD becomes an

administrative control device to selectively restrict host system access to the external domains (e.g. the Internet) and/or to confine signals incoming from external domains.

Referencing Fig. 3, a network IDS 10, as defined in Fig. 2, is shown protecting a set of internal domains 15, 15', etc. Members of a set of internal domains may include (networks, servers, PC's, laptops, etc.) a network IDS in the generalization of a standard IDS. The standard IDS protects one internal domain. Both type IDS systems are referred to in this document as an "IDS." The IDS 10 device includes programmable systems and includes an authentication processor 18 to implement a device-identification-number (DIN) authentication process to verify the identity and authorized presence of another IDS 10', or other device such as hosts 15, 15' in the network. The IDS 10' device includes an authentication process 18'. The communications subsystem of an IDS can use a DIN in the same manner that people use a PIN (personal identification number), with a bank card. DIN equipped IDS devices can operate a hardware-level inter-device authentication process. This DIN authentication process is operated during the initial handshake and randomly during a communications session, between IDS devices and/or other DIN equipped devices. A DIN can be variable, for added rigor. This process permits authorized network nodes/stations to identify any unauthorized and/or possibly malfunctioning nodes in a network. The IDS uniquely implements this process at the signal level of a network. Further, the DIN is encyphered by its IDS, for secure transit to other IDS devices. This encypherment can be accomplished using pattern representation of a

DIN. Such patterns will be unintelligible to unauthorized receivers. Authorized receivers can implement a pattern interpretation process (ARM '87) and thus gain core "adaptive capability" that would automatically enhance the DIN process. Also some limitations of binary computation are then removed from the DIN process. Thus, the process is invisible to hackers and other disruptors who operate at the software levels of a network. Additionally, the DIN authentication process can include device-status information on the IDS device identified by the particular DIN. This status information permits each IDS device (in a network of IDS devices) to both authenticate and react to the condition of other member-devices in an IDS network. This provides a degree of automatic fault-tolerance for an IDS network. In the network shown, host 15 is connected to IDS 10 through outgoing socket 17 and incoming socket 13 while IDS 10 is connected to the external domain 16 through socket 14 and to networked IDS 10'. IDS 10' is connected to the external domain, or another external domain, through socket 14' and through host input socket 13' to host 15' and socket 17' from host 15'. IDS device 10 and 10' illustrate a network of IDS devices (i.e., an IDS network). The DIN process, operating at the signal level enables the IDS devices 10, 10' to form a secure private network which is a network overlay to the external domain 16.

Referring to Fig. 4, the IDS can simultaneously protect multiple internal domains. Some or all of the protected domains can be remotely located from the IDS. The IDS architecture can utilize video teleconferencing technology. In this embodiment, an IDS 20 is defined, utilizing desktop video conference (DVC)

technology. As a brief background, operational interface standards for DVC are evolving. Generally the standard designations are as follows:

- H. 320→ DVC over the ISDN/POTS telephone environment
- H. 323→ DVC over LAN environment
- T. 120→ Collaborative Computing (e.g. Whiteboarding)

The majority of present DVC capabilities address either H.320 (telephone domain) or H.323 (LAN domain) either (or both) of which is the external domain 26 from which signals are received by an IDS 20. We now consider a DVC capability which addresses both the LAN and the telephone domains. Such a capability will permit simultaneous LAN and telephone domain connection. Conceivably, a user could connect to a classified LAN, and the Internet, simultaneously. Most InfoSec policies would forbid such simultaneous connectivity. In Fig. 4, a LAN/phone capable DVC device such as a PictureTel 550 is used in an IDS 20. The IDD 22 of the invention contains a LAN/phone DVC card. Generally, the DVC card is a peripheral-device to the system containing it. The DVC card also is obviously an external (interface) socket 24 for the IDS. A video-switch 23a is used to pass information to internal (protected) domains 25 and 25'. This switch is thus a socket to the internal domains. Each internal domain communicates with the IDS 20, in a remote-control DVC mode through receiver sockets 23 and 23'. This can be achieved by a simple "whiteboarding function" which is a standard feature, that can permit one computer system to control another. Specifics would be driven by the T.120 standard and the

particular devices used for implementation. By the video teleconferencing process, the information or original data set carried by signals from the external domain is processed through the IDD DVC card 24 so that the original data set is, at the output, a second data set from which information is extracted and is sent to the host domain in a video format. This conforms to the modified-read requirements for IDS operation. For applications where the unauthorized leakage/exfiltration of internal data, is of major concern, it should be remembered that the IDS 20 architecture via socket 27 forces all outbound signals from internal domains into the IDD 22. Signals in the IDD can be reviewed, manually and/or automatically for authorization, prior to interaction with external signals. This is a form of insider control. The IDS permits components to be remotely located. Also, the IDS can be remotely connected to its host system, with no reduction in the IDS ability to protect the host system. The IDS architecture is modular and thus permits modular maintenance and modular upgrade without adverse impact on the protection capability. As an example, for IDS applications using video signals, an advanced tv-card/video-signal-receiver can detect and filter unauthorized and/or undesired data signals imbedded in a video, e.g. tv signal transmissions. Such video receivers will, in their IDS function, isolate all incoming transmissions from program execution domains of the protected host system.

Referencing Fig. 5, a fundamental modified-read (M-R) process is illustrated. The modified-read operation deals with information transfer. Possibly contaminated signals and the data they carry are received from an external domain 37 via the

extended interface socket 34 of an IDS 30. In this example, the transfer is between a control module 31 and an external-interface-module (EIM) 32 of the IDS 30 which is, for example, a single board computer (SBC), embedded microprocessor (EMB) or embedded micro-controller (EMC) personal computer. The bus control signals from the EIM are restricted so that an EIM cannot, relative to the main IDS bus 33, become bus master and thus initiate data transfer. This is accomplished by disabling (e.g. grounding) the appropriate main IDS bus/(IDD internal communications segment) control signals from the EIM's internal interface.

The modified-read operation functions as follows:

IDS Control Module (CM) 31 scans the external request buffer of EIM 32 and checks request pending flag (note: EIM main memory contents must remain in the EIM, to confine possible contamination). If a request is pending, set read flag in the execution buffer file (EBF) 35. EIM 32 continually scans for read flag in EBF 35.

If read flag is set, the modified-read process is initiated to process the incoming signal from the external domain such as by a facsimile process, a conversion to video format process, or a printed format process.

When the modified-read sequence is complete, EBF 35 ready flag is set and the control module 31 transfers EBF 35 to main memory, for processing.

The above sequence defines the information transfer within a modified-read operation. The actual external data, which may be contaminated, never leaves the EIM 32. Information in the EBF 35 is transferred through socket 36 to the protected domain 38.

From the command of the control module, the EIM 32 will transfer its main memory contents to the probe memory (or holding area) in the CM 31. Subsequent steps are as follows:

Probe functions of the CM 31 builds an execution buffer file (EBF) 35. This is a coded representation of relevant (to the IDS function) contents of the EIM's main memory. This EBF 35 is what is actually transferred from the EIM 32 into the control module 31 of the IDS, for insertion into the IDD-to-internal domain socket 36. This process acts as an electronic air-gap, blocking the transfer of possibly contaminated data.

The IDD 40 via the CM 31 acts on the EBF 35. The EBF format and contents are unknown to external domains 37, and inaccessible from these domains.

The EBF is transferred to the protected domain 38 via socket 36.

The CM 31 returns status, response to requests, flush commands, etc. to the EIM. Actual CM 31 responses are obviously application specific.

The EBF, constructed by the EIM probe function, must conform to a proper set of EBF patterns/sequences authorized and recognized by the CM.

Contaminated external data never leaves the EIM 32. This condition is enforced by allowing no raw external data to leave the EIM, in-bound to a protected system 38.

A prime modified-read (M-R) objective is to prevent inadvertent or externally controlled execution of hostile code. Secondary objectives include forcing internal user deliberate interaction for execution of received external executable code. The following guidelines should be used for M-R implementation:

Incoming binary (including executable) data strings must:

- a) be modified to an alternate binary (non-executable) format;
- b) be treated as non-executable data (e.g. text data) by the receiving system; and
- c) be transformed, preserving information, but alternating data strings.

Incoming data stream (binary) must not re-appear in the system (without direct user action).

Transformation properties (at receivers) must:

- a) be known to external data transmitter;
- b) not have an inverse derivable by transmitter (thus eliminating cryptography); and
- c) map data stream into machine usable format.

By way of Example:

Take binary data stream; 1000111010010100001111010111 = (d_b)

Transformation $T_i|_{i \in \mathbb{N}^+}$

Then:

for example; $\dots f_i(0), f_i(1) \dots = T_i(d_b)$

$T_i(d_b); T_i T_i^{-1} \neq I$ no inverse exists (where I is an identity transformation)

$T_i(d_b) \neq (d_b)$ no unity, (for all i)

$T_i(d_b)$ is processable **only** in non-executable domains of the

receiving system.

By way of example, the modified read process may include the use of a facsimile machine to receive the incoming signal which may contain hostile data. The signal from the external domain is converted to print data which is a non-executable format at the receiving domain. The facsimile signals are scanned in, including by software, and forced into non-executable format for receiving domain processing.

The two primary InfoSec issues are first that possibly contaminated raw data does not enter the protected domain. Second, the incoming bit stream, the data virtual carrier, is not reproduced inside the protected domain. This second requirement is addressed by not using a direct inverse of the sending facsimile transformation. The information extraction transformation must not be an inverse of this original facsimile transformation. For some applications, an additional but not necessary safeguard would be restricting external knowledge of the actual recovery transformation used for the protected domain. If we view the original facsimile transformation as the transport transformation, and the scanning or print formation function as the recovery transformation, the general examples following could serve as transport/recovery transformation pairs:

- * EBCDIC/ASCII
- * $\text{Font}_i / \text{font}_j$

- * Fax_i / Fax_j (where Fax_j ≠ Fax_i)
- * text format/video format
- * text format/printer format
- * digital/analog
- * digital format_i / digital format_j (where digital_i ≠ digital_j)
- * signal format_i / signal format_j (where signal format_i ≠ signal format_j)

The "Hamming Distance" between the bit representation of one character, in the transport transformation, to its equivalent representation in the recovery transformation could, in some instance, serve as a measure of appropriateness for transformation pairs. Obviously, other transformation pairs and acceptability metrics could be derived.

The IDS process permits necessary information exchange between host computer systems and an external network without intrusion of (possibly corrupted) external data signals into the host. The modified-read process is a universal virus, worm, hostile executable code eliminator. This signal level, modified-read process operates below the software layer of a system. Thus, the process is not dependant on prior knowledge of hostile data structures (unlike conventional software-based anti-virus type packages) to neutralize such hostile data. This neutralization function is a primary host protection mechanism used by the IDS.

Referring to Fig. 6, a television signal based version IDS 42 is disclosed. The host-system 45 is a Packard Bell PLT 2240 personal computer system. The external-

domain 46 is the Internet/world-wide-web. Any PC or network of PC's can be protected in this manner. The intermediate domain device (IDD) 47 is a webtv system, for example Phillips/Magnavox MAT960A1 Internet Unit. The IDS 42 permits commercial off the shelf components to be used in their normal expected usage scenarios, without modification of any kind. As further illustration of this point, a television (PCI bus) card 48 (for example a Hauppauge 401 card) of the host system is connected to the webtv system unit. These are signal transformation processes that are implemented for the required modified-read process of the IDS. Such processes isolate all incoming signals from program execution domains of the host system, while making the "information content" of the incoming signals available to the host system 45. InfoSec integrity of the host is thus maintained.

As shown in Fig. 6, the tv card 48 transforms the output of the IDD 47 to a format different from that of the external domain 46 and which is processable by the host 45. Also shown in the drawing figure is an actual television 49 which is connected to an input of the television card 48 and which is utilized to verify that a true television signal is being received at the card thus insuring the correct operation of the tv card. As opposed to sending a signal from the webtv 47 to the television card 48, other signal transformations are possible, for example the signal can be outputted to a facsimile machine or printer 41 from the webtv IDD 47. The printer constitutes a signal transformation processor which preserves the information in a printed format as received from the webtv IDD 47. The preserved transformed signals of the print copy from the printer 41 can be scanned by a scanner 44 to

create a transformed signal which can be provided to the host system 45. A standard telephone 43 is also shown in the drawing figures and is utilized to check operation of the communications link between the IDS 42 (including the webtv system 47) and the external domain 46.

With continued reference to Fig. 6, the invention may also be used to protect the host during the updating of host system files. As shown, the host 45 may be connected at socket 50 such that files from the host can be downloaded to the IDD 47 of the IDS 42. In this embodiment, (which excludes use of a webtv type IDD) the file information is retained in a file buffer in the IDD. The IDD receives signals from the external domain and processes the signals as described in Fig. 5 with respect to IDD 40 to thereby perform the modified-read process and obtain signals having a different data set. Information is extracted from the initial data set in such a manner as to derive a second data set which is then sent to the file buffer to update the file information downloaded from the host 45 and the updated file is thereafter forwarded as a tv signal to the socket or tv card 48 of the host. Thus, the file of the host is updated without any undesirable data being transmitted to the host system. In some embodiments no host to IDD socket exists. Thus, no signal path for exfiltration of the domain signals is available. With the protected system thus isolated from cyberspace and/or other hostile domains, it can be safely connected to a classified domain/network without danger of compromise to that classified domain.

The intermediate domain system of the present invention is a system within a

system type architecture wherein such systems and subsystems may be activated and deactivated to achieve maximum IDS functional flexibility. As an example, if the IDS is implemented to reside internal to his host, the host interface module is activated. If the IDS is implemented to reside external to the host, a communication subsystem linked to the host/internal domain is used to embody an outgoing socket between the protected host and the IDS similar to socket 17 of Fig. 2. In either case, the modified-read subsystem includes the incoming socket from the external domain.

With reference to Fig. 7, the IDS operation will be described in detail. A data set, possibly contaminated, is received by the communication subsystem where it is important to note that the data set is carried in a signal format as previously discussed and the signal format may also be corrupted. The processing data flow controller subsystem accesses the received data set and determines if it is program and/or control data that must be processed. If program execution is required, the data set is transferred to the external processing domain (of the IDD) for execution and the results of the execution are returned to the processing data flow controller subsystem for transfer to the modified-read subsystem. The IDS provides the necessary platform on which to execute efficient, secure procedures for handling external executable programs. An example of such procedures is discussed below. If no program execution is required, the processing data flow controller subsystem transfers the data set to the modified-read subsystem directly. The modified-read (M-R) subsystem operates as described with respect to the embodiment of Fig. 5

discussed above. The "reset" function is then executed, as described with respect to the embodiment of Fig. 2, also discussed above. It is also important to note here, that the protected system can be a component of the IDS. An example is a secondary mass storage system that receives information processed by the (M-R) process.

Fig. 8 illustrates a multifunction IDS configured for video teleconferencing. The IDS chassis 51 is that of its host such as 45 of Fig. 6, if the IDS is implemented to reside internal to its host. In this case, all add-in cards of Fig. 8 (i.e. cards 52a, 52d 52b, 52c, and 54; whereby card 52a is a modified single board computer (SBC) and card 52d is a video capture card, card 52b is a graphics accelerator, 52c is a sound card, card 54 is a modem type embodying an external domain interface socket. The socket may be in the form of a modem board or a network or cable interface type card. The cards 52a, 52b, and 52c comprise the intermediate-domain-device (IDD) of the IDS. As shown, an IDS can reside internal to its host, if its SBC's interface to the host's expansion bus is configured as an add-in card. The SBC 52a thus uses only devices directly connected to it, and not those devices connected to the host's expansion bus. For the case of an IDS implemented to reside external to its host, the add-in cards reside on the passive backplane of the IDS chassis 51. The SBC 52a implementing the control module of the IDD, controls the IDS from its slot on the IDS device's passive backplane. Cards 53 and 53a form a socket, and are a tv card 53 and a sound card 53a both residing in the host system's expansion bus. Socket 57 is a one way direct cable connect (DCC) link from the host system to the SBC and is

used for direct data transfers to the IDD. Modules 31, 32, 33 and 35 (from Fig. 5) reside in the SBC 52a. The internal hard drive 62 is connected to the IDD's SBC 52a and resides in a bay in the chassis 51 of the IDS or, the chassis of the host, if the IDS resides internal to the host. A compact-disk (CD) drive 63, backup tape drive 64, floppy disk drive 65, and the smart-card reader 66 can each reside internal to or external to the chassis 51, where each device is connected to the IDD's SBC 52a, permitting the IDS to operate as an independent system whether residing internal to or external to its host. A joystick 67 as well as a microphone 68 are connected to the IDD sound card 52c, to support telephony, video telephony, network gaming, and video conferencing type functions. In addition to its InfoSec functions (and those just mentioned), the IDS is ideal as a special function platform, which frees the host for simultaneous execution of other tasks. Video monitor (VGA) signals 69, move from 52a to 52b to socket 53. Audio signals 70 move from 52c to socket 53a. This video and audio information transfer is a "video based" modified-read process. Signals 72 and 73 are video and audio output from the host domain. Signals 71 from a keyboard or mouse 75 are applied to the IDD's SBC 52a. Finally, a video camera 74 necessary for video conferencing and video telephony operations is connected to the card 52d of the IDS. Using the teachings of the invention, all incoming signals from all input sources such as to the modem 54 which receives signals from the external domain 80, the camera 74, disk drive 63, tape drive 64, floppy disk drive 65, smart card reader 66, joy stick 67 and microphone 68, are processed through the cards 52a, 52b, 52c, 52d acting as the IDD and are transformed so that the

host/protected domain remains safe and isolated from the external signal source, which may be contaminated. If a desktop-video-conferencing (DVC) type card is used for an input socket 54, instead of a standard modem, microphone and video camera inputs could go directly to the DVC card. A V.90 standard (or better) compatible modem is recommended for older telephone system type video-phone usage. Other, high bandwidth, high performance modems and other communication type devices such as network interface cards, cable system interface devices may be used to embody socket 54. All external signals, contaminated or not, are confined within the IDD.

Referencing Fig. 9, an SBC 120 to be used is an IDD residing internal to its' host, such as the computer 45 of Fig. 6, must be modified in the manner of Fig. 9 wherein the bus control and arbitration signals 122 are deactivated such as by grounding at 123 and the bus master/slave signals 131, 132 and 133 are enabled such that the modified SBC 120 interfaces to the host peripheral bus 128 as a standard add-in card. The PCI bus specification is used in Fig. 9 to illustrate this generic modification procedure. The modified SBC retains its' internal/on-board connections 130, 130' and 130'' to which SBC dependent peripheral devices may be connected thus forming a "system within a system" capability for the host.

When modifying the SBC for use in a "system within a system" environment, the following procedures must be followed:

a) the SBC arbitration-control signals must be disabled to prevent control arbitration of the protected systems expansion-bus by the SBC;

b) enabling only the bus-master and bus target capability of the modified SBC which respectively permits initiating and reception of expansion-bus data set transfers; and

c) ensuring that the interface to the protected system's expansion bus can not act as a bridge module between the protected system's expansion bus and the IDD device's on-board local bus, thus isolating on-board bus connected devices from the protected system's expansion-bus connected devices and enabling a secure "system within system" architecture. These generic modifications are embodied in an expansion-bus interface module (EBIM) for an SBC. An EBIM is obviously specified to the expansion-bus architecture of the host/protected system.

The three generic modifications discussed above are achieved for example when the protected systems expansion bus conforms to the peripheral component interconnect (PCI) bus 128 by allowing the modified SBC add-in card 120 functioning as the IDD to assert an REQ# (a bus request) at 131 and to only receive GNT# (bus grant) control signals 132, and ACK# (acknowledge) type signals 133 in a PCI configuration, thus ensuring the IDD peripheral devices are not directly accessible from the protected system's expansion bus. A multiplicity of such modified SBC systems can be used in a single IDD, to render that IDD extremely fault-tolerant, and dynamically flexible. Such modification would be included in an EIBM embodiment for a PCI expansion-bus.

Referencing Fig. 10, the traffic flow control feature of (all versions of) the invention is illustrated, that will apply to both incoming and outgoing data signal

traffic. The illustrated feature uses an adaptive process. An example of such a process is that presented in "An Adaptive Reference Monitor for Information System Security," by C. Kellum; IEEE Proceedings, 1st Annual ICNN, '87.

Security flow control and dissemination control are IDS functions that are usually required for routers, servers, and gateways, however optional for user-workstations. The IDS controls signal traffic flow to and from its protected system. This includes (counter-flooding type functions) which control:

- type of traffic flow
- rate of traffic flow
- priority of traffic flow
- determination of originating point of traffic flow

These "application specific" functions are intrinsic capabilities of the IDS architecture. Also, they are primary IDS functions in protecting network devices such as hubs, switches and routers. Thus, a "message flooding" effect of contaminates can be effectively reduced or eliminated, since the IDS is an optimum point, in a network, for security flow control.

Generally viruses, worms, and mutations thereof, would be intrinsically and automatically eliminated after one encounter with an IDS. Thus a network of IDS protected systems, (a secured physical private network (SPPN) for example), would automatically eliminate contamination without any danger to protected systems, or specific user operations. It is important to note that the term "contamination" is meant to include all forms of hostile type code. In addition to virus and worm type

code, privacy violating type contaminates (such as "cookies"), corruption of operational code (for example, by external system error), and other forms of malicious and/or corrupted code are included in the term "contamination". Thus, an IDS protected system or network is also isolated from contamination that could result from external Y2K type problems.

Determination of the point of origin of incoming signal traffic is a useful tool in retro-active countermeasures. Such may include counterattack on (and/or isolation of) the launching sites of hostile incoming signal traffic. A network of IDS devices which includes a DIN process has the ability to determine points-of-origin of any signal traffic within that IDS network. This determination process can be invisible to the systems being protected by the IDS network, since the DIN authentication process is a normal communication process in such an IDS network. Further, all IDS devices (in such a network) which interface directly to an external domain, can act as launching sites for probes into the external domain with the purpose point-of-origin determination. As an example, commercial network-trace type programs (such as tracert DDS/Windows or traceroute (UNIX)) can be launched from an IDS. Thus, depending on the particular application involved, an IDS or an IDS network can serve both in a target identification platform and counterattack launch platform. The IDS can also serve as a platform for advanced procedures, necessary to achieve secure system operation. A primary example is procedures for securely handling "externally generated" external executable programs, some of which may be contaminated.

Some InfoSec problems cannot be handled by technology alone. New procedures must accompany new technology (in many cases), if a safe information processing environment is to be achieved.

A prime "generic" example of this is the use of externally generated executable code. Examples are spreadsheets, word macros (e.g. open), equations in symbolic processor packages (e.g. Mathematica, etc.), JAVA, HTML, etc. Today, these type "programs" are used to process sensitive internal data. These programs can be contaminated, either by error and/or by malfeasance. Macro viruses are an example of such contamination.

Firewalls CANNOT stop these macro-viruses, some are attached to legitimate e-mail, which passes through a firewall. Such contaminated programs/macros can compromise important internal information with which it comes into contact. New viruses, or other forms of contamination, can be generated at any time. Thus, anti-virus software is generally non-effective with new contamination (e.g. the Melissa worm). Also note that a proxy-server is just an application level firewall.

To safely utilize external executables, one must have the proper hardware in place which enables proper InfoSec procedures to be followed. One cannot reliably defend against software (SW) with other SW. The proper hardware (e.g. an IDS) must prevent any external signals, including executable, from entering a protected domain. All external signals, including executables, macros, spreadsheets, etc., must be confined in the IDS. Then the following generic operational procedure can be followed (with application-specific fine tuning):

internal data selected for processing is moved to the IDS;

external executables operate on their data (in the IDS), thus any contamination is confined in the IDS;

results of the processing operation are sent to the internal domain via the M-R process; and

the IDS is reset, thus removing any contamination.

It should be noted that "External executables" can be saved to an IDS mass storage device to avoid loss of the "external executable" under IDS reset. This is not recommended if there is possibility that the "external executable" code is contaminated. Further, an IDS can utilize removable mass storage device. Such devices can be physically "off-line" during external session, thus preventing contamination. Contents of devices remaining "on-line" will undergo a "modified-read process" prior to use by protected systems. To insure against undetected contamination of "external executables," their point of origin and content should be authenticated. This can be accomplished by the IDS (Device Identification Number processing) capability.

IP-spoofing/man-in-middle attacks can be defended against by proper use of the DIN processing capability, as previously noted, attack countermeasures can authenticate point-of-origin and data-unit content in an IDS network. Such countermeasures incorporate DIN processing capability of the IDS. Examples of implementation options for DIN processing in an IDS include:

PCMCIA;

smart-cards;

MSBC; and

embedded μ -controllers.

A DIN equipped IDS uses the DIN processing capability to authenticate a device (or set of devices) to which it is connected. A DIN can serve as a private key in an asymmetric cypher environment. Two IDS devices can exchange an authentication message, each encyphered in the recipient's public key. The decode of the authentication message insures an authorized connection, because an intruder would not have the secret key with which to decode the authentication message. The DIN/"secret-keys" do not appear on the network. The public-key can be viewed as a Device Permission Number (DPN). Each site has a list of DPN's for sites which it is authorized to connect with. The DPN list can be dynamic, if warranted by a specific application.

Example:

Site A

Site B

1. $\rightarrow Q(k_p^B, T) \rightarrow$
2. Site-B decyphers $Q(k_p^B, T) \cdot k_s^B$
(B now has T, in plain text)⁵
3. $\leftarrow Q(k_p^A, (T + \Delta t)) \leftarrow$
4. Site-A decyphers $Q(k_p^A, (T + \Delta t)) \cdot k_s^A$
(A now has $(T + \Delta t)$, in plain text)
5. Site-A and Site-B are now a mutually authenticated transceiver pair
6. session is set up, optional session keys issued or message encyphered in receiver's public key

Key:

$T \rightarrow$ authentication msg.

$k_p^i \rightarrow$ public key (of i^{th} site)

$k_s^i \rightarrow$ secret key (of i^{th} site)

$Q \rightarrow$ asymmetric cypher function

$\Delta t \rightarrow$ time stamp + ack msg.

As noted, DIN's can be implemented on smart-cards or PCMCIA cards and an authentication processor can be implemented on MSBC, PCMCIA card, smart card, or some combination.

As keys do not appear on a network, hackers/man-in-middle do not have key access

and thus cannot intercept, alter, or disrupt messages, without detection. The use of an authentication/cypher co-processor in the IDS can be embodied in several ways ranging from an MSBC to PCMCIA cards to embedded μ -controller type systems. The processing power makes asymmetric cyphers for full data units practical by combining asymmetric and symmetric cypher processes in the authentication/cypher co-processor.

DIN processing capability used to authenticate data origin and content forms the "label processing" basis for multi-level secure networking. The DIN capability gives labels to IDS sites. DIN based subnetworks can be static or dynamically allocated "locally" by an individual IDS operating on its authorized DIN-list. The DIN-list indicates the sites with which a respective IDS may communicate. A non-IDS equipped site is given an application specific "null-DIN." Since an IDS is always contaminate-free during a session initiation cycle, due to the IDS reset function, the DIN processing function is un-corrupted. The DIN process extended to data-units in a network of IDS devices, provides a device (point-of-origin) and data content authentication capability for a network of IDS devices handling data-units on that network. Thus, via the DIN processing capability, sites/nodes on an IDS network can monitor other sites on that network. This capability is a basis for a "2 site rule" mechanism, whereby an insider at one site must have permission from another site (or sites), to perform designated actions. Thus the term "2 site rule". In the above asymmetric cypher example, data-units would be encyphered in the public key (k_p^i) of the originating site (e.g. site i). The DIN processing function is a form of label processing, thus a platform for multi-level-secure (MLS) operation, on an IDS network, is provided. The IDS is a platform for trustworthy key management. Thus an IDS device may contain several keys

(i.e. a set of keys). Key selection can be accomplished in the manner of commercial encryption packages hosted on the IDS, or by IDS-unique processes where keys are inserted/activated using smart-cards, PCMCIA type cards, or like devices. Such keys, used for combined crypto-sealing and authentication of data units, make the data units unavailable to sites without the proper key or set of keys. This is a basis for multi-level secure (MLS) operation on an IDS protected network.

Referencing Fig. 11, an example protected system 140 is connected via an IDS 141 to both a public telephone network 144 (such as the Internet) and an internal/corporate LAN (local area network) 145. The LAN consists of several IDS protected sites/nodes 141, 141_{i,j,k,l} and some unprotected nodes 146, 146_{φ,λ}. Each IDS 141 has a removable cypher-key 147, for DIN processing. Thus, connections between IDS protected sites can be activated at the level of the inserted cypher-keys. Consequently, a physical connection between IDS protected sites can be activated at different levels, depending on the cypher-keys used at the connecting sites. This forms a basis for multi-level-secure (MLS) internetworking. Devices 141_m and 141_n are respectively modem and NIC (network interface card) devices.

Referencing figure 11, each IDS protected site is isolated by the IDS protecting that site. The isolation is achieved by the modified-read (M-R) data path 142, and the outbound filter path 143. If 143 is not used/eliminated, then no (malicious or otherwise) output from the protected system 140 can take place. External 144 (e.g. Internet) or internal 140 generated contamination signals are confined to the IDS 141, and subsequently eliminated via the IDS reset function. Malicious internal system 140 generated signals that

pass the IDS filter function would only effect "non-IDS protected" sites. An IDS protected network is thus self-cleaning. If IDS protected sites are to share an Internet connection, the Internet-sharing hardware and software would reside in the IDS devices of the sharing sites. Thus, any commercial Internet-sharing package can be used in an IDS protected environment.

Fig. 12 illustrates a generic architecture for a system-on-chip (SOC) version of the invention. The external interface module 150 is the interface between the SOC and the outside world (e.g. the Internet). Module 150 can range in type from a wire socket (such as for a telephone cable jack) to a microstrip type antenna for wireless tranceiving. The processor modules 151 provide the computation power for the SOC. Components 152 are reconfigurable-logic-unit (RLU) devices, and are commercially available. An RLU is an SRAM (static random access memory) based programmable-logic-device (PLD). The RLU devices are used in lieu of conventional bus type components, as they can provide dynamic-reconfiguration and fault-tolerant (F-T) capabilities. Application-specific-integrated-circuit (ASIC) devices 153, 153' are used to provide functional flexibility for the SOC version of the invention. The RLU devices 152 permit selective activation of the ASIC components 153, 153', and the processor components 151. Also protected system interface components 154, 154', 154'' are selectively activated using the RLU 152. Additionally, the processor components provide the domains for safely processing external executable code (which may be contaminated), and confining that code (and its processing results) within the SOC. The reset line 155 is used to initialize the components it is connected to. The reset function insures the SOC returns to a non-contaminated, secure state (e.g. its initial state). Depending on the application being served by the (SOC), the reset function can be

executed pre-session, post-session, and/or manually. The SOC architecture accommodates mixed-signal components, InfoSec specific signal control (such as the modified-read function), and operational enhancement via the ASIC modules.

Fig. 13 illustrates an example (SOC) implementation of the invention. This example shows relevant InfoSec related components. The top portion of figure 13 shows the incoming *signal screen* function. Incoming wireless signals are received by the antenna 160, which is connected to the communications sub-system's receiver module 161. Processor 162 is the input signal processor. Processor 163 is a bank of general purpose processor modules utilized by the screen function of the invention. The RLU 164 selectively (according to the given application) connects ASIC modules 165, 165', 165'', 165''', 166. In this example, 165''' is the display memory of the video subsystem of the invention. The output of the ASIC modules is selectively connected to the modified-read module 167 and the video based modified-read module 167', via the RLU 164'. The output of the modified-read modules 167, 167' are selectively connected (via RLU devices 164'') to the protected system interface modules 168, 168'. Modules 168, 168' generate a carrier signal and pass the information received from 167, 167' to the protected system. The modified-read component 167''' is an RF type mixed-signal component. This process insures isolation of the protected system from any external signals.

The bottom half of figure 13 illustrates an outbound (from the protected system) filter/guard (i.e. signal flow-control) function. Component 170 is the interface to the protected system's outbound data socket. Component 171 generates a one-way optical signal 171' which is received (in a telemetry/broadcast mode) by the outbound processor 172. This is a mixed-signal function on the outbound circuit. RLU 164 selectively connects

ASIC modules 173 with the processor 172. Results of the filter/guard processing are sent to the output processor 175 via the RLU 164. The transmitter module 161' transmits (via antenna 160'). The reset line 180 is used for both the outbound and inbound halves of the SOC version of the invention, in this example.

It is expected that the present invention and many of its attendant advantages will be understood from the forgoing description and it will be apparent that various changes may be made in form, implementation, and arrangement of the components, systems, and subsystems thereof without departing from the spirit and scope of the invention or sacrificing all of its material advantages, the forms hereinbefore described being merely preferred or exemplary embodiments thereof.

The foregoing description of the preferred embodiment of the invention has been presented to illustrate the principles of the invention and not to limit the invention to the particular embodiment illustrated. It is intended that the scope of the invention be defined by all of the embodiments encompassed within the following claims and their equivalents.